National Institute of Technology Karnataka, Surathkal

Department of Computer Science and Engineering

CS254 - Database Systems Lab

Project Report

Champions Arena

A Competitive Programming Platform

Submitted by:

Aditya Chandrashekhar Sohoni (181CO203)

Manas Trivedi (181CO231)

Index

Introduction

Competitive programming is generally involved in the first round of most IT companies' campus recruitment procedures all over India. To prepare for the same, students learn competitive programming and sharpen their skills in the same on numerous online platforms. These platforms organise contests which inspire and encourage friendly competition among the students. At the end of the contest, each participant appears on the leader board based on the number of problems solved. There is also a social aspect to such platforms. Users can make connections with other users and view their profiles.

This project, 'Champions Arena', is an attempt to create a similar platform where programming enthusiasts can try, solve, and learn from a wide variety of problems to improve their logical and critical thinking. In particular, this project aims to study the database design involved in the creation of such platforms.

We chose this project as both of us are passionate competitive coders, and wanted to do something that we could relate to.

'Champions Arena' is a Java Application, which we have developed using NetBeans IDE 8.2 and the MySQL RDBMS. In the IDE, we used Java for front-end designing and for providing functionalities to the UI components, and MySQL for the creation and management of the database.

To run the project, follow the instructions given in the README here.

The following is the link to the GitHub repository of the project:
https://github.com/manasdtrivedi/Champions-Arena

Existing System

As of 18th June 2020, the application has the following functionalities:

1. Sign up and login of users, handled by the USER table.
2. Searching users based on User ID (uid), name and/or college, handled by the USER table.
3. Sending, unsending, and accepting connection requests to/from other users, handled by the CONNECTION table.
4. Viewing profiles of other users, who are present either in search results, have sent connection requests, are friends, or are present in the leader board (current/final standings of contests), handled primarily by the USER table, along with other tables.
5. Viewing problems, either from previous or current contests, handled primarily by the PROBLEM table.
6. Viewing current and previous contests, handled by the CONTEST table.
7. Viewing the contest standings of each contest, handled by the SUBMISSION table.
8. Submitting and receiving verdict for code submissions, handled jointly by the PROBLEM and TESTCASE tables.
9. Receiving a position on the contest standings upon submitting a problem on a current contest, handled by the SUBMISSION table.

Proposed System (Future Extensions)

The following functionalities may be added to the existing system as extensions:

1. Viewing editorials of problems of a contest, after the completion of the contest.
2. Global rating of participants, based on a weighted average of performance in previous contests.
3. Profile updation and deletion.
4. Forums for discussions.

System Overview

The tables that were created and used in this project are:

1. USER
   uid (User ID)
   uname (Name)
   passwd (Password)
   college (College)

2. CONNECTION
   user1 (Sender of connection request)
   user2 (Receiver of request)
   status (0 - Receiver is yet to accept request, 1 - Receiver has accepted)

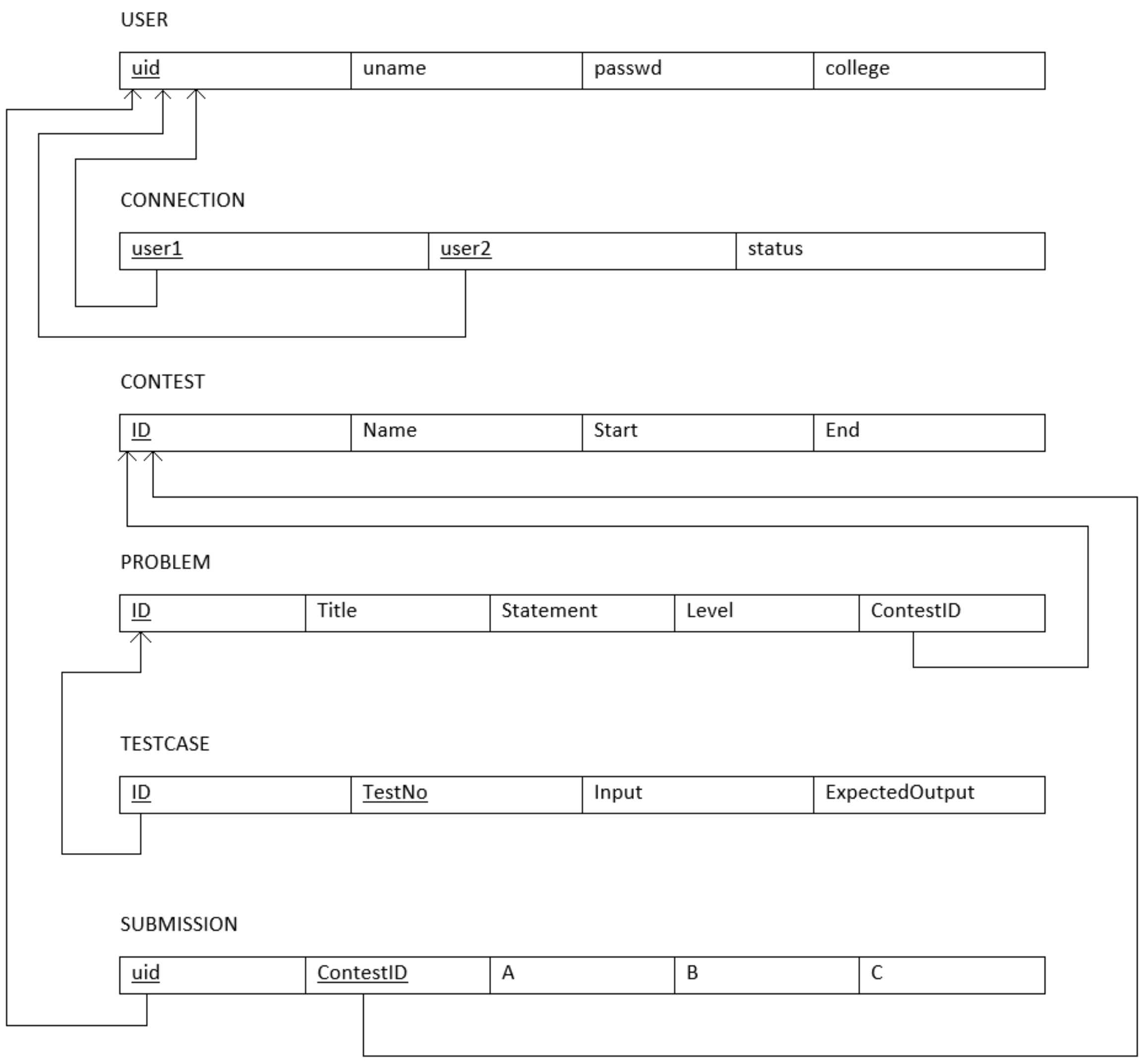
3. CONTEST
   ID (ID of contest)
   Name (Name of the contest)
   Start (Start date and time of the contest)
   End (End date and time of the contest)

4. PROBLEM
   ID (ID of problem)
   Title (Title of the problem)
   Statement (Statement of the problem)
   Level (Level of the problem)
   ContestID (ID of contest to which problem belongs)

5. TESTCASE
   ID (ID of problem for which testcase is written)
   TestNo (Test Case Number)
   Input (Input for user's code)
   ExpectedOutput (Output to be compared with user code's output)

6. SUBMISSION
   uid (User ID of user)
   ContestID (Contest ID of contest user has taken part in)
   A (A bit that is set if the user has solved problem A of contest)
   B (A bit that is set if the user has solved problem B of contest)
   C (A bit that is set if the user has solved problem C of contest)

The ChampionsArena Database has the following schema:



Relationships between entity types:

1. CONNECTS_TO
   This is an M:N relationship between USER (user1) and USER (user2)

2. SOLVED_BY
   This is an M:N relationship between USER and PROBLEM.

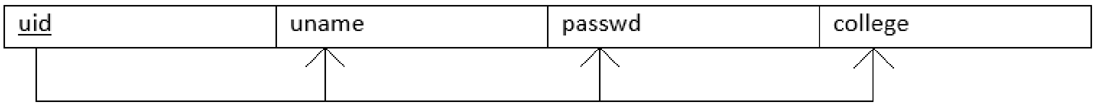
3. PARTICIPATED
   This is an M:N relationship between USER and CONTEST.

4. ARE_IN
   This is an N:1 relationship between PROBLEM and CONTEST.

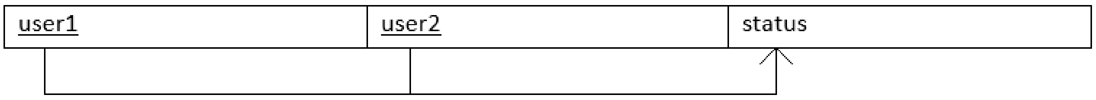
5. HAVE
   This is a 1:N relationship between PROBLEM and TESTCASE.

Normalisation

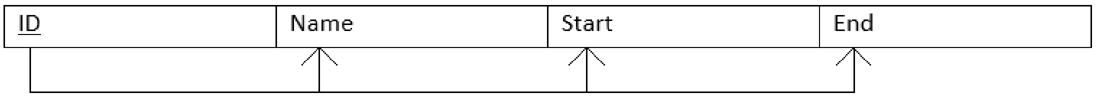
USER



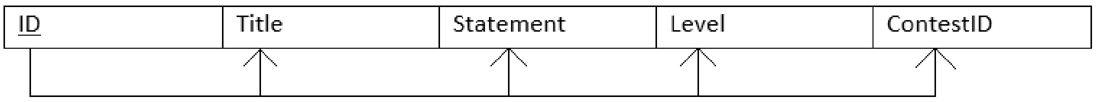
CONNECTION



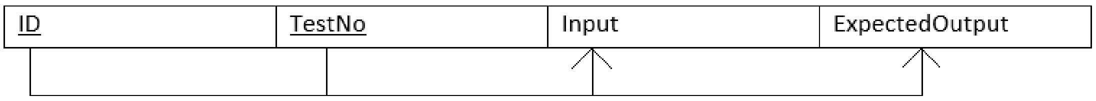
CONTEST



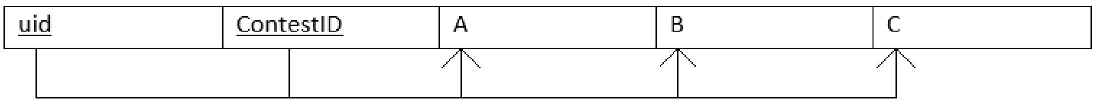
PROBLEM



TESTCASE



SUBMISSION



The following functional dependencies are present in the database:

1. uid → uname passwd college
2. user1 user2 → status
3. ID → Name Start End
4. ID → Title Statement Level ContestID
5. ID TestNo → Input ExpectedOutput
6. uid ContestID → A B C

First Normal Form:

The database does not have composite attributes, multivalued attributes and nested relations. Hence, the database is in 1NF.

Second Normal Form:

Each non-prime attribute is fully functionally dependent on the Primary Key of its relation. Hence, the database is in 2NF.

Third Normal Form:

In each functional dependency in the database, the left-hand side is the Primary Key of the relation on which the functional dependency is defined. Hence, the database is in 3NF.

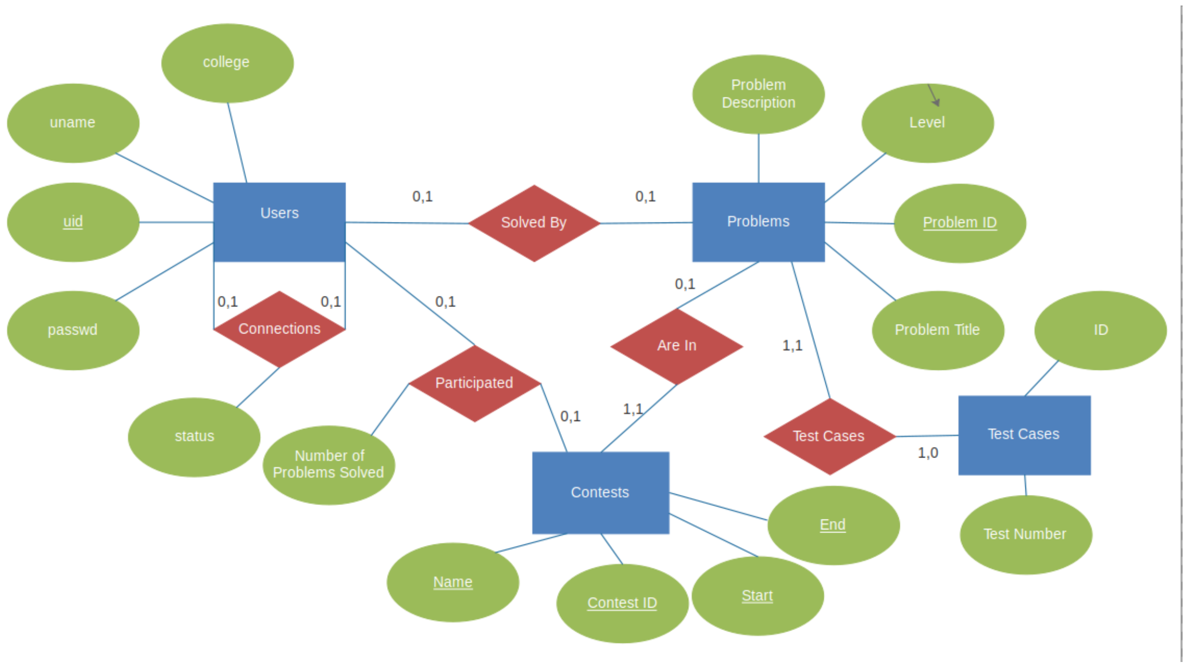
Boyce-Codd Normal Form:

By the same reasoning as that for 3NF, the database is also in BCNF.

Fourth Normal Form:

In each functional dependency in the database, the left-hand side is the Primary Key of the database. Hence, for the same value of the left-hand side, there cannot exist more than one tuple in the relation, which means that multiple values of RHS for the same LHS aren’t possible. This means there are no multivalued dependencies. Hence, the database is in 4NF.

ER Diagram



This is a screenshot of the ER Diagram that was generated by using the EDrawMax software. The .eddx can be found in the Github repository.

Implementation (Coding)

The application was created using NetBeans IDE 8.2 in Java, with MySQL as the RDBMS.

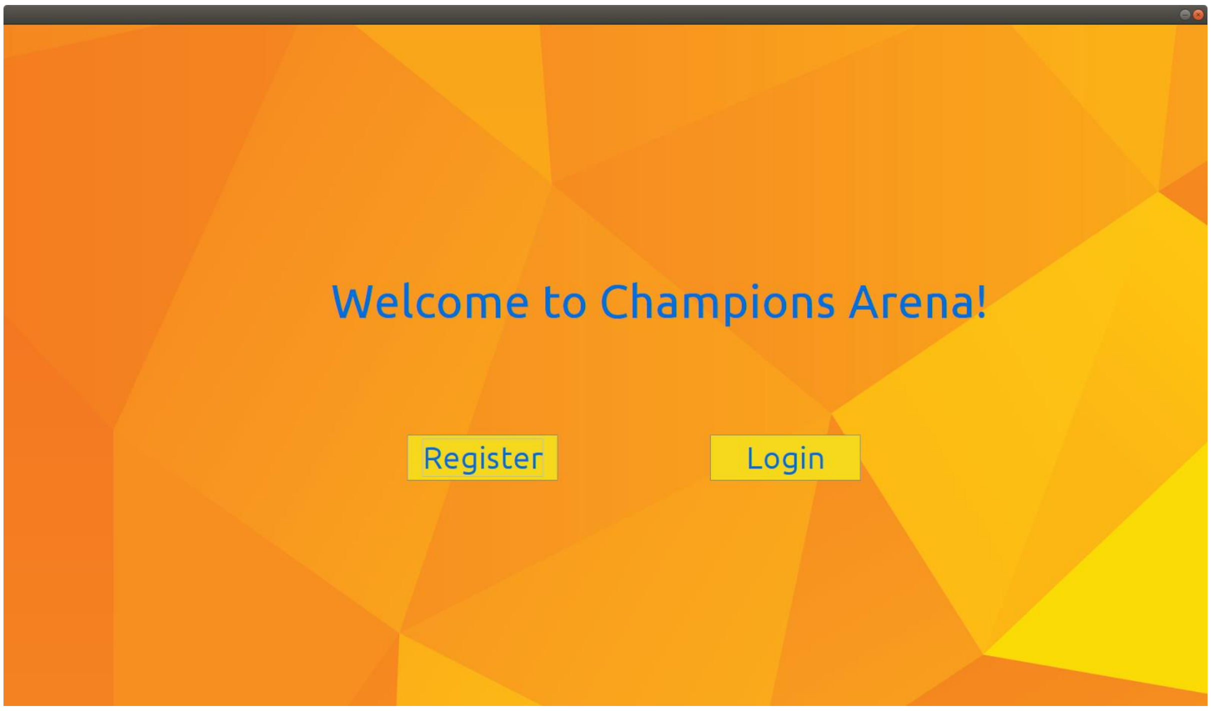
The user interface was designed using the Rapid Application Development (RAD) tools offered by the IDE. The code to be run upon events being triggered, such as the clicking of a button, opening of a window, clicking ‘Enter’ in a textfield, etc., was written in Java.

To run SQL statements, first a connection between the Java application and the database in MySQL is established. Then the SQL statements are sent to MySQL, and the result obtained upon execution is either displayed in a table, or operated upon to generate further results. This is done by using the DriverManager, Connection, Statement, and ResultSet classes in Java.

The main code of the application is present here.

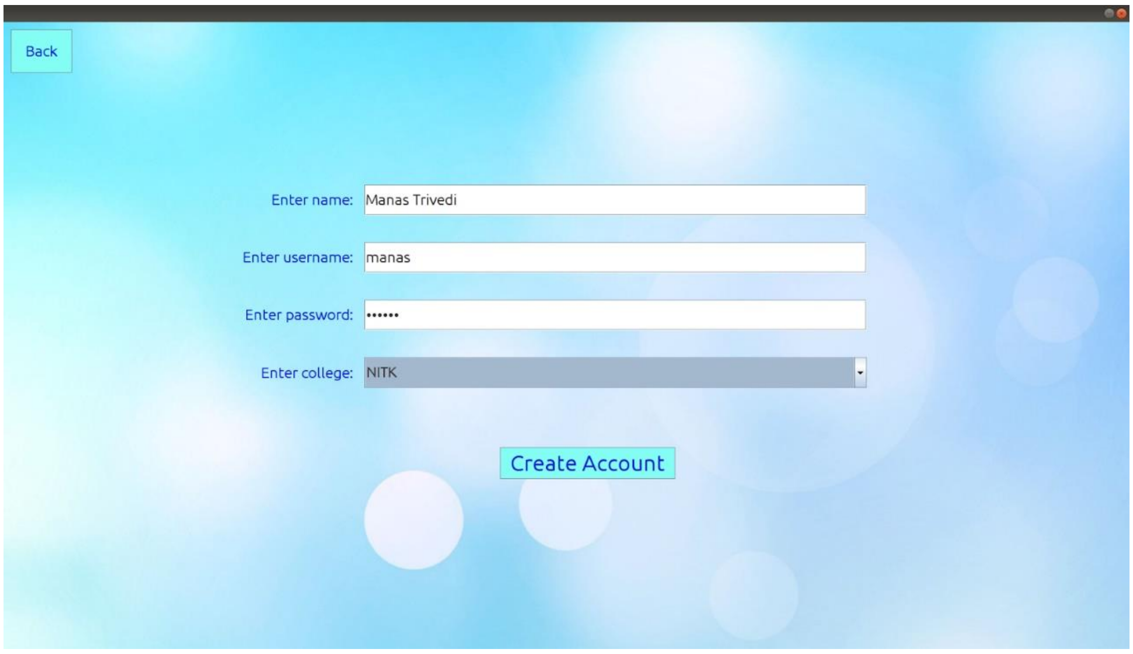
The application contains 12 frames (or windows/pages).

The Home Page



This is the first frame to open when the application is executed. The Register button takes the user to the Register frame, while the Login button takes the user to the Login frame.

Register Frame



Enter name: Manas Trivedi

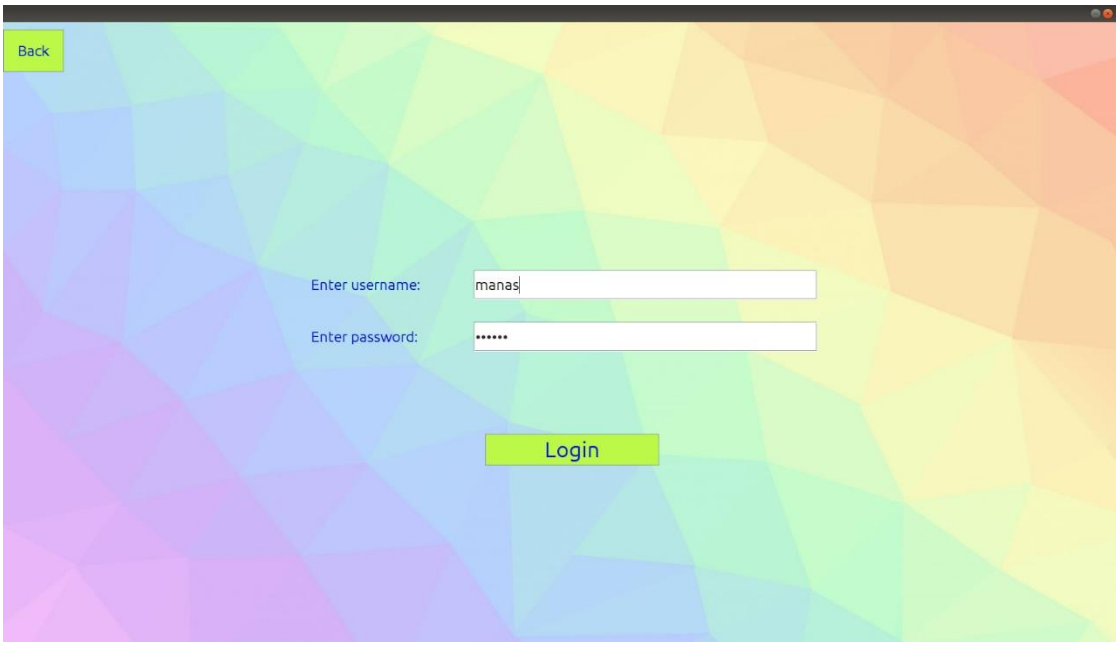
Enter username: manas

Enter password: ••••••

Enter college: NITK

In this frame, the user creates an account, by giving details such as name, username, password and college. If the username has already been taken by another user, a message dialog gets displayed which asks the user to enter another username. Also, if the college of the user is not present in the college combo box, then an input dialog gets displayed where the user can enter his/her college.

Login frame

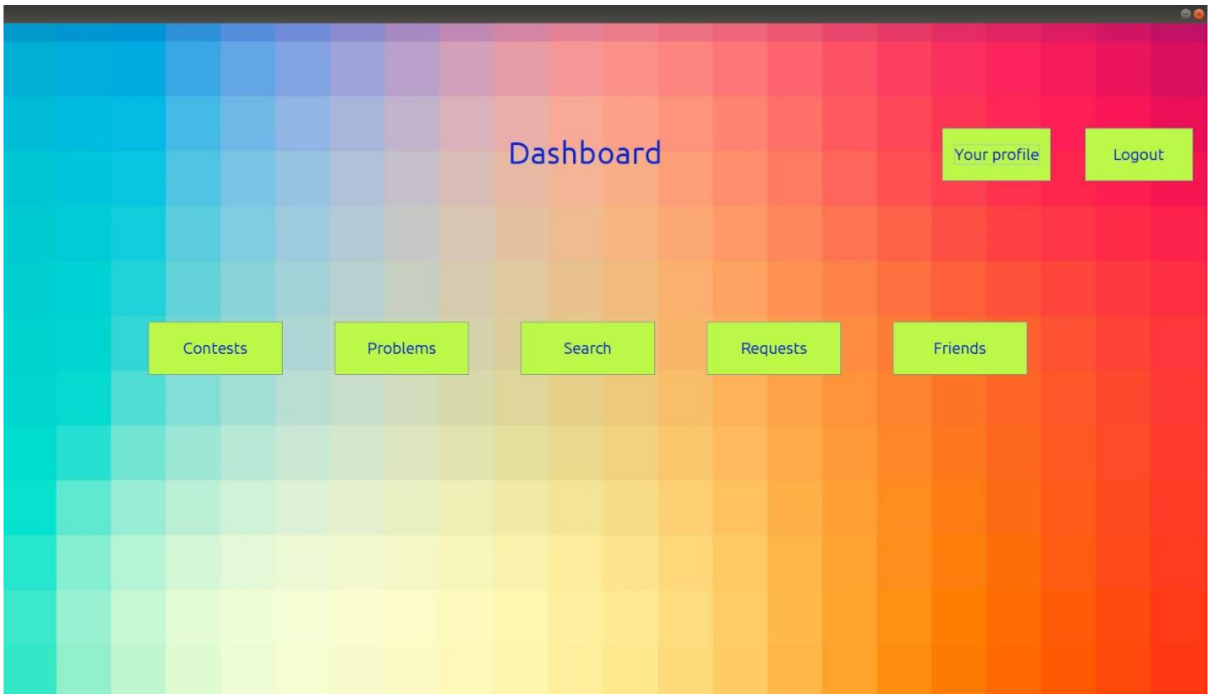


Enter username: manas

Enter password: ••••••

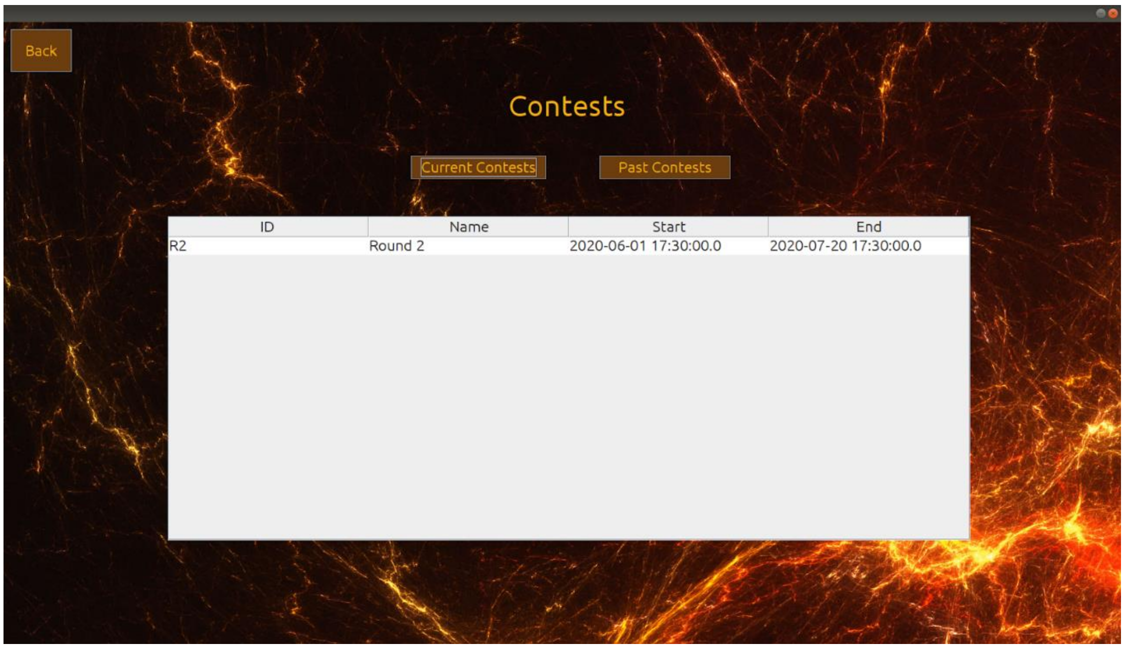
Here, the user enters his/her username and password. Based on the username entered, the correct password is fetched from the database and is compared with the entered password. If the passwords match, then the user is taken to the Dashboard frame. Otherwise, a message is displayed asking the user to enter the right password.

Dashboard Frame



This is the frame from where most of the other frames are accessed. By clicking the Contests button, the user is taken to the Contests frame. Clicking the Problems button leads the user to the Problems frame, where problems from all previous contests are displayed and can be solved, but without any leaderboard membership. The Search button opens up the Search frame. The Requests button takes the user to the Requests frame. The Friends button directs the user to the Friends frame. The Your Profile button directs the user to the Profile frame, where he/she can view his/her own profile. And finally, the Logout button takes the user back to the Login frame.

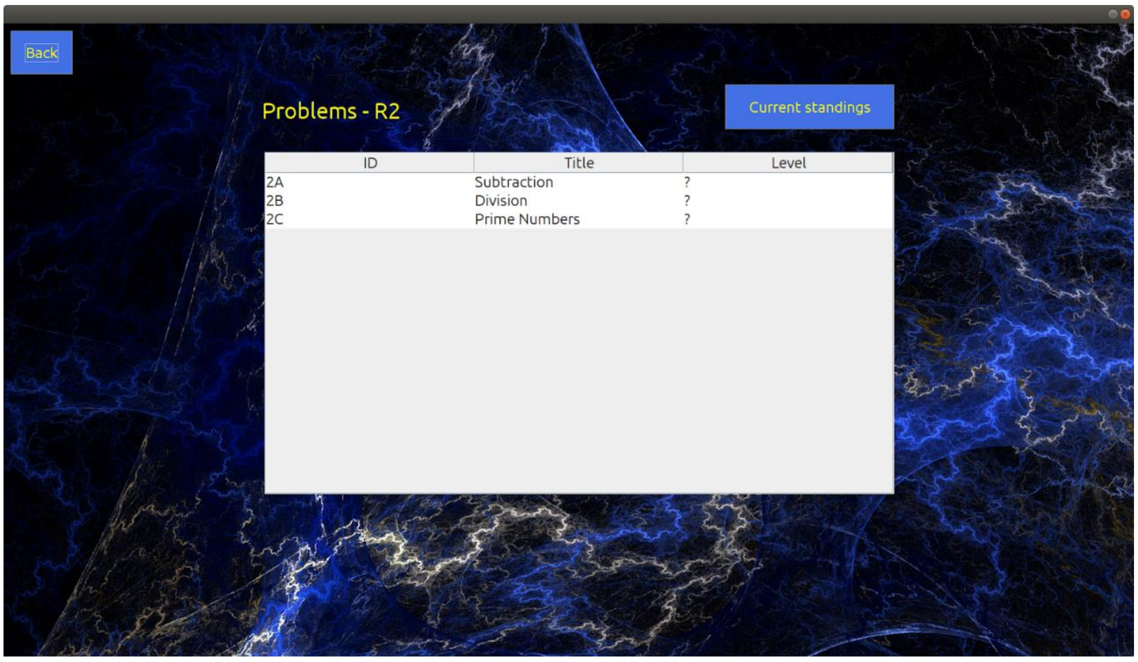
Contests Frame



ID	Name	Start	End
R2	Round 2	2020-06-01 17:30:00.0	2020-07-20 17:30:00.0

Here, initially all Current Contests (the contests whose Start time is less than or equal to the current time, and End time is greater than or equal to the current time) are displayed. The user can view the Past/Current Contests by clicking on their respective buttons. In the table, the user can click the contest he/she wishes to solve, and will get directed to the Problems frame, where problems from the selected contest are displayed.

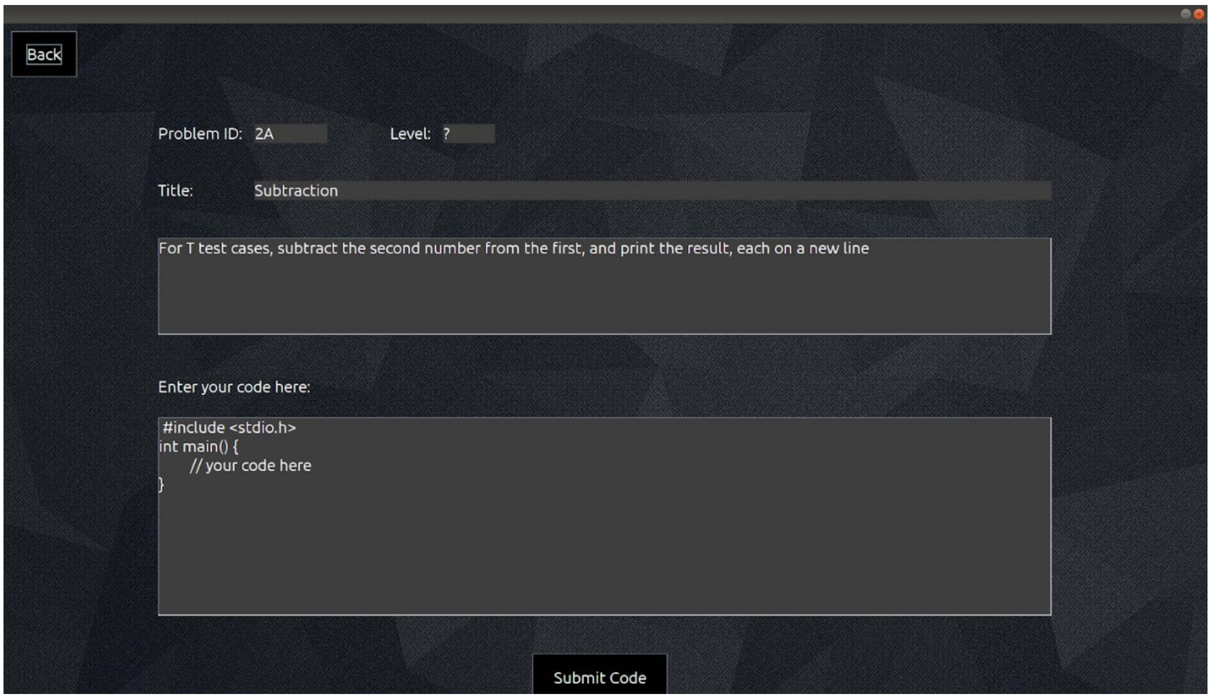
Problems Frame



ID	Title	Level
2A	Subtraction	?
2B	Division	?
2C	Prime Numbers	?

This frame opens up when the user clicks on a contest in the Contest Frame, or clicks the Problems button on the Dashboard. Accordingly, the Current Standings button on this frame displays Current/Final Standings or is hidden. If the problems displayed are for a contest, then the Current/Final Standings button will take the user to the Standings frame. If the user clicks a problem on this frame, he/she is directed to the Solve Problem Frame.

Solve Problem Frame



Problem ID: 2A

Level: ?

Title: Subtraction

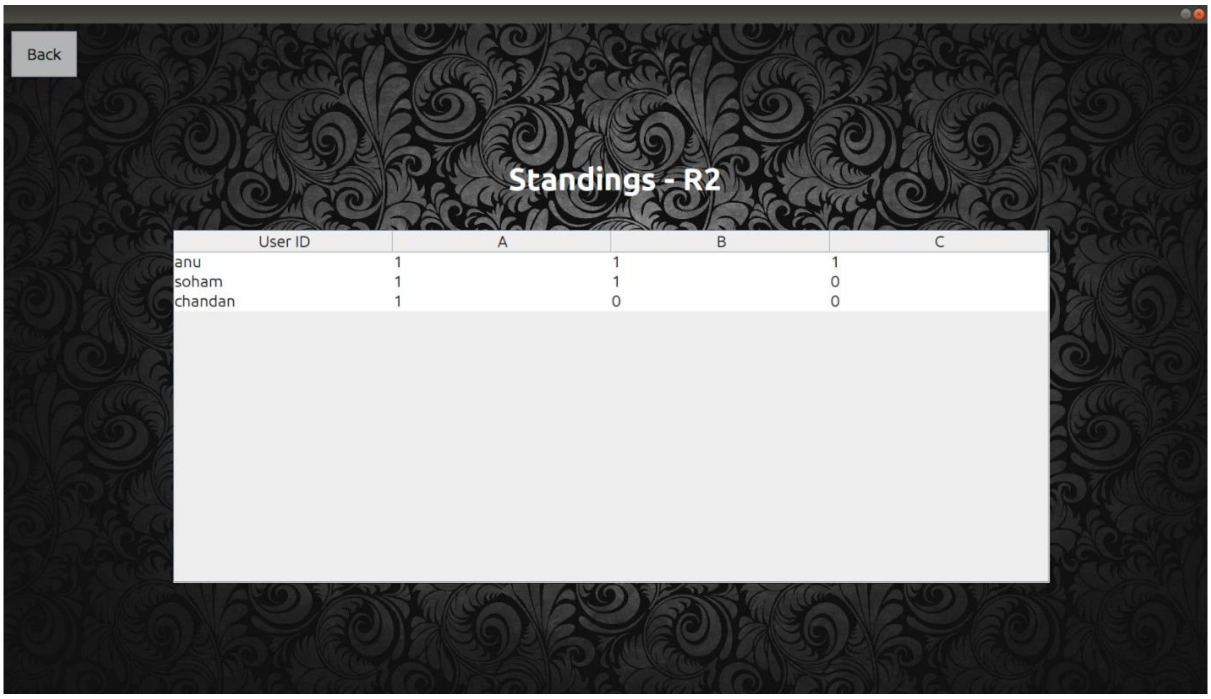
For T test cases, subtract the second number from the first, and print the result, each on a new line

Enter your code here:

```
#include <stdio.h>
int main() {
      // your code here
}
```

This frame is where users submit code for a particular problem and get a ‘Compilation Error’ / ’Correct!’ / ‘Wrong answer. Try again!’ verdict. If the user submits code for a problem present in a current contest, then the record for the user is updated, or inserted if not already present, based on the verdict. The code is compiled, and is checked for compilation errors. If successfully compiled, the executable file generated is run on multiple testcases, where each testcase consists of inputs and corresponding expected outputs. The compiled code is run on these inputs, and the outputs produced are compared with the expected outputs. If the outputs are correct, then a ‘correct’ verdict is given, and ‘wrong answer’ otherwise.

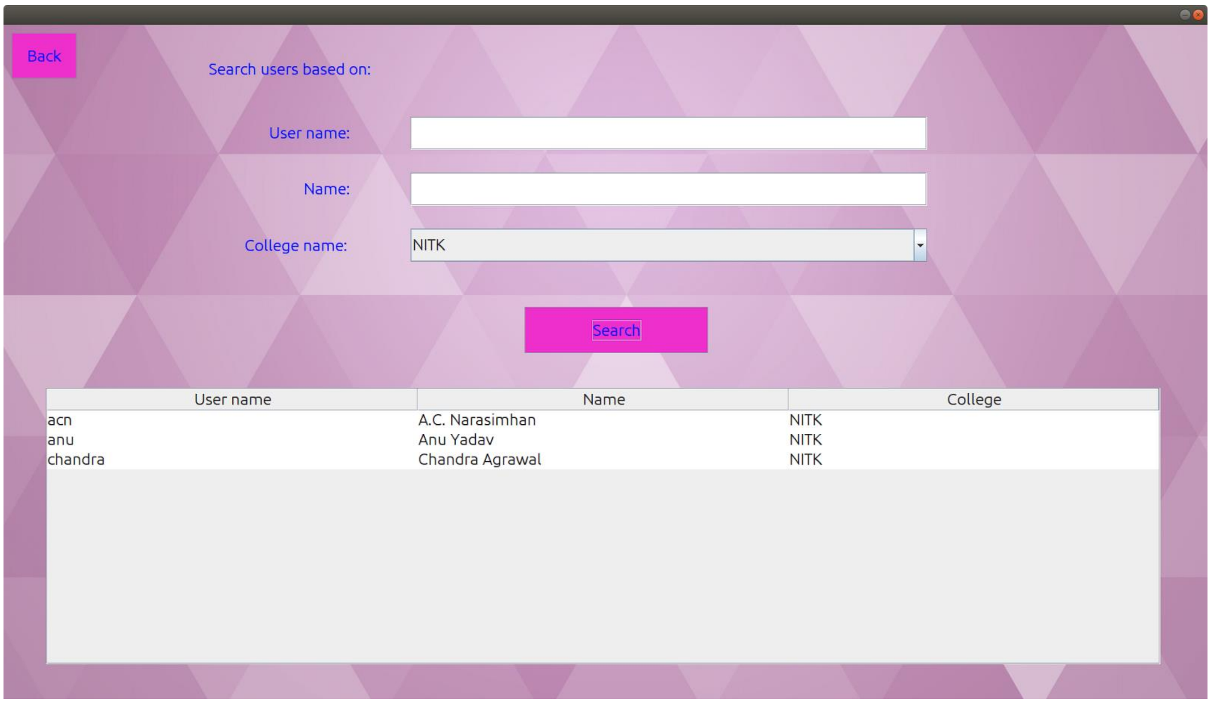
Standings Frame



User ID	A	B	C
anu	1	1	1
soham	1	1	0
chandan	1	0	0

This frame displays the current/final standings (leaderboard) for current/previous contests. If the user clicks a user's record in this table, he/she is directed to that user's profile.

The Requests Frame



User name:

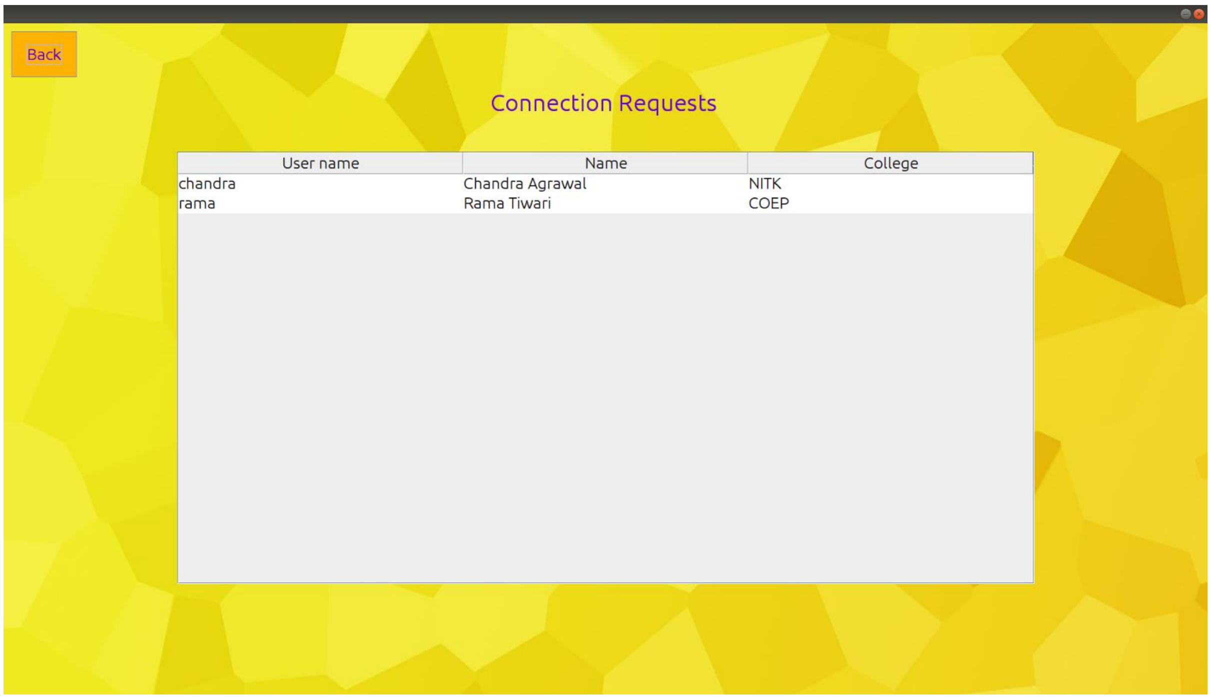
Name:

College name: NITK

User name	Name	College
acn	A.C. Narasimhan	NITK
anu	Anu Yadav	NITK
chandra	Chandra Agrawal	NITK

In this frame, the user can search for other users, using filters of username, name and college name. If the user decides not to apply any filters, then the records of all users (except the current user) are displayed. Clicking a user's record in the above table takes the user to the selected user's profile.

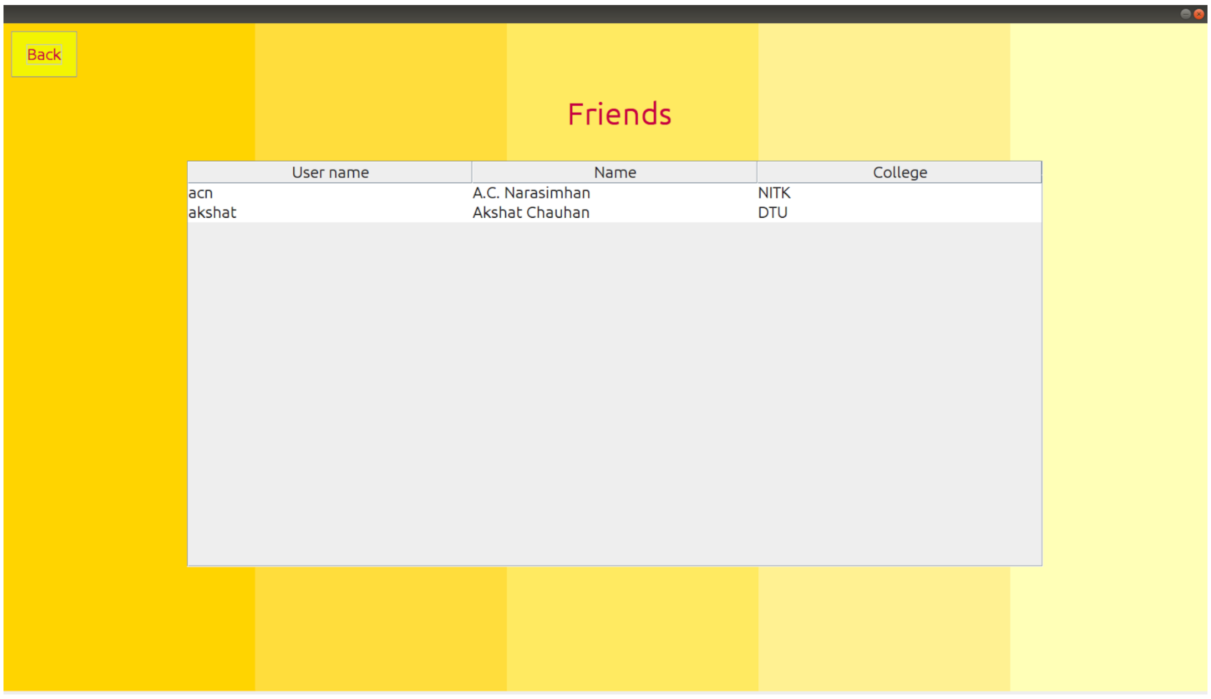
Requests Frame



User name	Name	College
chandra	Chandra Agrawal	NITK
rama	Rama Tiwari	COEP

Here, the user can view the records of the users who have sent connection requests to him/her. Upon clicking the user's record in this table, the user is directed to the profile of the selected user, where the user can click the 'Accept Request' button to accept the connection request.

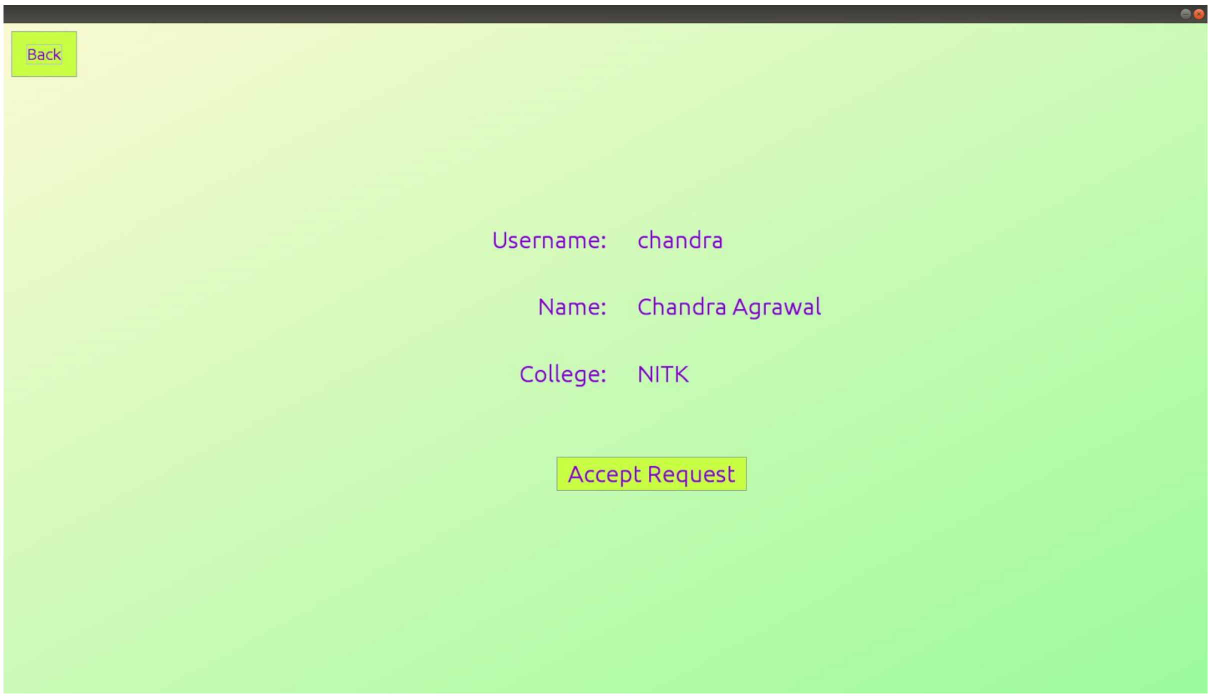
Friends Frame



User name	Name	College
acn	A.C. Narasimhan	NITK
akshat	Akshat Chauhan	DTU

Here, the user can view the users with whom he/she is a friend with i.e. the users who accepted his/her connection request, and the users whose connection requests were accepted by him/her.

Profile Frame



Username:	chandra
Name:	Chandra Agrawal
College:	NITK

This frame appears when the user clicks the ‘Your profile’ button on the Dashboard, or clicks a user’s record in any of the tables which appear in the application. In the former case, the user will be able to see his/her own username, name and college. In the latter case, the details of the user whose record was selected will be displayed. If this selected user has sent the current user a connection request, then the ‘Accept Request’ button will be displayed. Else if the selected user isn’t a friend, then a ‘Send/Unsend Request’ button will be displayed, based on whether the current user has sent a request. And finally, if the selected user is already a friend, then no such button will be displayed.

Conclusion

During the course of this project, we learnt many useful skills of database management, and application design. We made sure to keep user friendliness and easy understandability as top criteria while making both the project as well as the report.

We would in the future like to improve on this project, and add ratings and a forum for competitors. Most of all, we would like to practically deploy this project so that many beginners can learn about competitive programming, using a sleek user-interface, elegant design, and easy to follow instructions, all that have gone into the making of “Champions Arena”.

References

Arora, S. [2017] *Informatics Practices*, 10th ed., Dhanpat Rai & Co., 2017

Elmasri, R., and Navathe, S. [2007] *Fundamentals of Database Systems*, 5th Ed., Pearson, 2007